

Weekly Report

基于排名学习的地图匹配方法

摘要: 从全球定位系统(Global Positioning System, GPS)获取的地理位置往往和实际的地理位置有一定的偏差。这制约了基于地理数据的数据挖掘、可视化、智能交通系统等方向的研究与应用。地图匹配是解决这一问题的方法之一,它将从GPS获取的地理位置匹配到对应的道路中。之前的地图匹配方法从几何、拓扑、概率等方面分析数据特点,建立数学模型来完成地图匹配任务。这些模型只能较好地反映研究时所用数据的特点而不能很好地反映具体应用场景的数据特点。这降低了这些方法的泛化性。为解决这一问题,本文提出了一种数据驱动的基于排序学习的地图匹配方法,可从数据中学习地图匹配规则。据我们所知,这是世界上首次提出的从数据中直接学习地图匹配规则的方法。实验证明,我们的方法在XXX方面的有XXX效果,并在运算速度上相对部分算法有显著优势。

关键词: 地图匹配; 排名学习; 神经网络; 可视化; 地理信息系统

A Map Matching Method Based on Learning to Rank

Abstract: Omitted.

Key words: Omitted;

(本次修改只修改了标题及摘要部分,引言部分暂未修改)

全球定位系统(Global Positioning System, GPS)是利用定位卫星,在全球范围内实时进行定位、导航的系统。近年来,顺应人们生活和工作的需要,手机等便携式计算设备有了飞速的发展。大多数设备装有GPS模块,使得人们更易于利用GPS数据实现诸如导航、重要目标定位等简单应用,另一方面也更容易获取海量的GPS数据。越来越多的GPS数据为研究交通拥堵,行程预测,道路关联性分析等交通问题提供了新的研究途径^[4]。然而,由于设备性能及安全方面的原因,往往无法获得准确、高频的GPS数据。由于卫星时间等参数的测量误差,实际获取的GPS数据往往和真实位置有1~20米的偏移;由于传输过程数据的丢失,设备的流量及功耗,实际的GPS数据采样频率通常是不稳定且存在大量低频数据的^[2]。低频数据带来了两采样点之间轨迹的不确定性。这一不确定性在一些与GPS数据相关的研究中又称为采样误差^[1]。

原始GPS点数据的测量误差及采样误差限制了很多基于地理位置的研究和应用。往往在使用这些数据前,需要对它们进行一些预处理。地图匹配(Map Matching)则是解决该问题的数据预处理过程之一。地图匹配针对目标的出行数据,假设获取的GPS数据都位于道路上,结合GPS数据和道路网络数据即可估计出目标的出行轨迹和每个GPS数据点所在的道路及其大致位置。地图匹配算法按照应用类型可分为在线(online)处理和后(offline)处理算法,按照所使用的技术类型可分为几何法(geometric),拓扑法(topological),概率法(probabilistic)和其它高级技术法(如隐马尔科夫模型HMM)。目前比较著名的后处理地图匹配算法是基于HMM的地图匹配算法^{[2], [4], [5]},这些算法针对稀疏有噪音的数据取得了比较好的效果。然而,这些算法大部分针对某些限定条件的情况,如文献^[4]的方法只适用于与参数有关的特定范围的采样频率。根据文献中的数据^[6],当单条轨迹的点数变多时,算法的耗时会剧增。此外文献中的方法^[1-7]大多根据交通专家对交通规则的理解,建立了一定的数学模型,如文献^[7]中认为采样点的速度与道路的平均速度越接近,则目标行驶在该道路上的概率越大。由于社会的发展,交通规律在迅速地演化,

专家的知识有一定的滞后性，未必符合目前的交通规律，从而造成泛化性的降低及匹配结果的错误。这使得交通数据分析领域对无模型的地图匹配算法提出了需求，如文献[8]则针对最短路径的假设，提出基于力牵引的方法解决。

针对现有方法存在的问题，我们引入神经网络，提出一种新的地图匹配方法。神经网络是一种经典的深度学习模型，目前在分类、检测、分割等图像处理领域成果显著，但将其应用在并非与图像处理直接相关的数据预处理领域尚有困难。我们提出的方法根据 GPS 轨迹数据和地图数据构造一条最接近实际路径的曲线，再根据这条曲线，用简单快速的地图匹配算法在地图中找到最接近的道路序列作为最终的匹配结果。我们采用计算机图形学中的样条曲线混合函数来描述轨迹的曲线段，利用卷积神经网络(convolutional neural networks)对点图像和地图图像分别做卷积逐级地做特征提取，将图像嵌入到高维空间中。最后在高维空间利用回归模型生成样条曲线的参数值来控制样条曲线去接近真实的轨迹。由于目前的地图匹配算法对高采样率的 GPS 数据表现较好，我们将高频数据下采样后作为训练数据，将高采样率 GPS 数据的地图匹配结果作为标注，以验证我们提出算法的有效性。

据我们所知，我们的工作首次将神经网络引入地图匹配的研究中。优点：略

本文的剩余部分按如下结构组织内容：略

1 相关工作

（介绍地图匹配领域与排名学习（learning to rank）领域的研究进展。）

1.1 地图匹配

（地图匹配研究进展。）

1.2 排名学习

（排名学习介绍。）

2 基于排名学习的地图匹配方法

（本节介绍我们提出的方法。）

2.1 数据及问题定义

（我们要解决的问题的 GPS 轨迹、路网、匹配结果、候选道路集、地图匹配等概念形式化的定义。）

2.2 网络结构

（输入、输出数据格式及网络的结构。）

2.3 损失函数

（训练所用损失函数及训练过程。）

3 实验结果及分析

（本节介绍验证我们方法有效性的实验。）

3.1 数据说明

（说明我们实验用到的数据。）

3.2 训练过程及评判标准

（说明我们的训练过程和评判标准。）

3.3 结果分析

(展示最终结果， 并做相应分析。)

4 结论

(总结全文。)

References :

[1] Hall T, Beecham S, Bowes D, Gray D, Counsell S. A systematic literature review on fault prediction performance in software engineering. IEEE Trans. on Software Engineering, 2012,38(6):1276–1304.

[2] Hunter, Timothy, Pieter Abbeel, and Alexandre Bayen. "The path inference filter: model-based low-latency map matching of probe vehicle data." IEEE Transactions on Intelligent Transportation Systems 15.2 (2014): 507-529.

[3] Zheng, Yu. "Trajectory data mining: an overview." ACM Transactions on Intelligent Systems and Technology (TIST) 6.3 (2015): 29.

[4] Lou, Yin, et al. "Map-matching for low-sampling-rate GPS trajectories." Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems. ACM, 2009.

[5] Newson, Paul, and John Krumm. "Hidden Markov map matching through noise and sparseness." Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems. ACM, 2009.

[6] Yuan, Jing, et al. "An interactive-voting based map matching algorithm." Proceedings of the 2010 Eleventh International Conference on Mobile Data Management. IEEE Computer Society, 2010.

[7] Hu, Gang, et al. "If-matching: Towards accurate map-matching with information fusion." IEEE Transactions on Knowledge and Data Engineering 29.1 (2017): 114-127.

[8] Rappos, Efstratios, Stephan Robert, and Philippe Cudré-Mauroux. "A force-directed approach for offline GPS trajectory map matching." Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. ACM, 2018.

工作时间

星期	任务	Duration
周一至周五、周日	调试代码，看文论，构思论文，写论文	9:00 - 12:00 和 14:00 - 22:30，共 11 小时

Work Time: above 50 hours